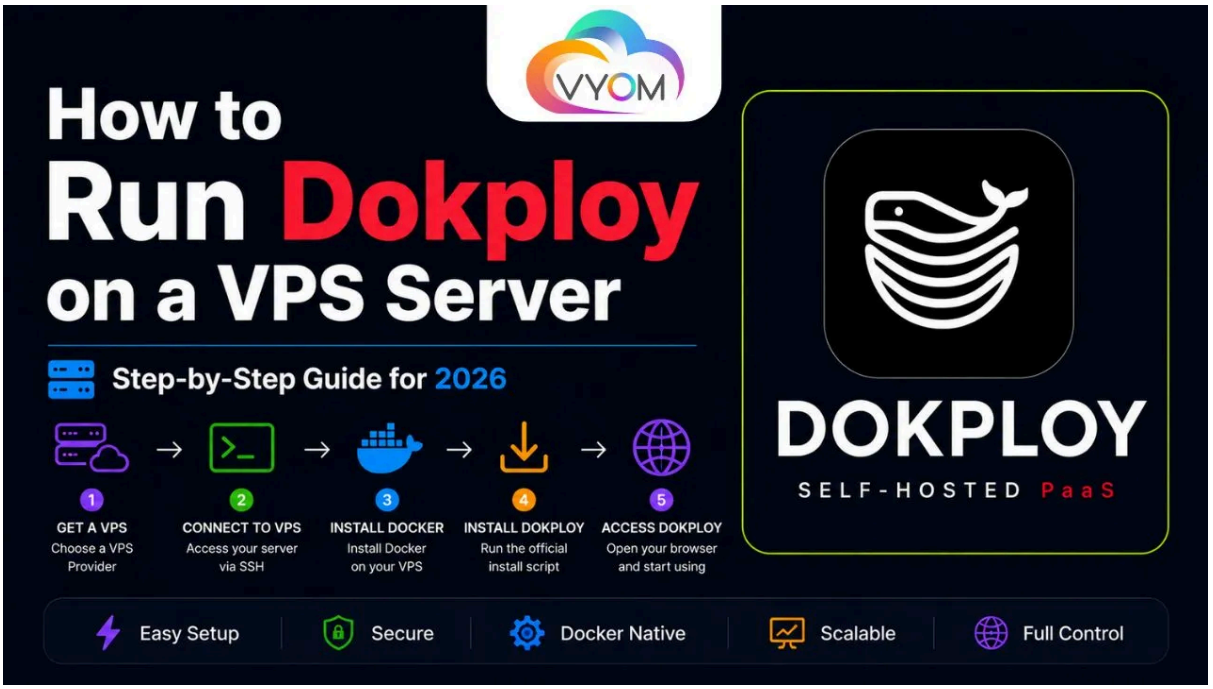
How to Run Dokploy on a VPS Server (Complete 2026 Guide)



Running your own applications on a Virtual Private Server (VPS) gives you complete control over your hosting environment while keeping costs low. Instead of relying on expensive managed cloud platforms, many developers now prefer self-hosted deployment solutions that provide flexibility, scalability, and ownership of their infrastructure.

Dokploy has quickly become one of the leading open-source deployment platforms for self-hosting applications. It offers a modern web dashboard, Docker-native deployment, Git integration, database management, and automatic SSL certificates—all from a single interface.

If you're planning to host personal projects, client websites, SaaS applications, or internal business tools, installing Dokploy on a VPS is an excellent choice. This guide explains everything you need to know, from choosing the right VPS to deploying your first application.

Why Use a VPS for Dokploy?

A VPS offers dedicated resources without the high costs of a dedicated server. It provides the flexibility to install and configure software according to your needs while maintaining full control over your environment.

Benefits of using a VPS with Dokploy include:

- Affordable monthly hosting
- Full root access
- Better performance than shared hosting
- Scalable infrastructure
- Complete server ownership
- Easy application deployment

Many developers choose VPS providers because they strike the perfect balance between cost and performance.

Minimum Server Requirements

Before installing Dokploy, ensure your VPS meets these basic requirements.

Minimum Specifications

- Ubuntu 22.04 LTS
- 2 vCPU
- 4 GB RAM
- 20 GB SSD storage
- Root or sudo access
- Stable internet connection

Recommended Specifications

For production workloads:

- 4 vCPU or higher
- 8 GB RAM
- SSD or NVMe storage
- Daily backups
- Dedicated IP address

These specifications allow Dokploy to run smoothly while leaving enough resources for hosted applications.

Choose a VPS Provider

Dokploy works with almost any cloud VPS provider.

Popular options include:

- DigitalOcean
- Hetzner
- Vultr
- Linode
- Amazon EC2
- Google Cloud
- Microsoft Azure

Choose a provider based on your budget, preferred data center location, and scalability requirements.

Step 1: Connect to Your VPS

Use SSH to access your server.

```
ssh root@your-server-ip
```

Replace **your-server-ip** with your VPS IP address.

After logging in, you are ready to prepare the server.

Step 2: Update the Operating System

Keeping the operating system updated improves security and compatibility.

Run:

```
sudo apt update && sudo apt upgrade -y
```

Once completed, reboot the server if required.

Step 3: Install Docker

Dokploy relies on Docker for running applications.

Install Docker using the official installation script.

```
curl -fsSL https://get.docker.com | sh
```

Verify Docker installation.

```
docker --version
```

You should see the installed Docker version displayed.

Step 4: Verify Docker Compose

Check whether Docker Compose is available.

```
docker compose version
```

Modern Docker installations include Docker Compose by default.

Step 5: Install Dokploy

Dokploy provides an automated installation script that configures everything required.

Run:

```
curl -sSL https://dokploy.com/install.sh | sh
```

The installer automatically:

- Downloads required components
- Creates Docker containers
- Configures networking
- Starts Dokploy services

Installation typically completes within a few minutes.

Step 6: Open the Dashboard

Once installation finishes, open your browser and visit:

```
http://your-server-ip:3000
```

You'll be prompted to create an administrator account.

Enter:

- Name
- Email address

- Secure password

After registration, you'll gain access to the Dokploy dashboard.

Step 7: Add Your VPS

Inside the dashboard, register your server.

Navigate to:

Servers → Add Server

Provide:

- Server name
- VPS IP address
- SSH credentials

Dokploy will verify the connection and prepare the server for deployments.

Step 8: Connect a Git Repository

Dokploy supports several Git providers, including:

- GitHub
- GitLab
- Bitbucket
- Gitea

Once connected, Dokploy can automatically deploy applications whenever new code is pushed to your repository.

This enables a streamlined continuous deployment workflow.

Step 9: Deploy Your First Application

Create a new application by selecting **New Application**.

Configure:

- Repository
- Branch
- Build settings
- Environment variables

Dokploy supports popular frameworks such as:

- Node.js

- React
- Next.js
- Laravel
- PHP
- Python
- Go
- Docker-based applications

Click **Deploy**, and Dokploy will handle the build and deployment process.

Configure Custom Domains

To make your application publicly accessible, add a custom domain.

Examples:

- app.example.com
- api.example.com

Update your DNS records to point to your VPS IP address.

Dokploy automatically detects the domain and prepares it for SSL configuration.

Enable HTTPS

Secure websites improve user trust and SEO performance.

Dokploy integrates with Let’s Encrypt to:

- Generate SSL certificates
- Configure HTTPS
- Automatically renew certificates

No manual certificate management is required.

Monitor Applications

Dokploy provides built-in monitoring features.

You can view:

- Deployment history
- Application logs
- Container logs
- Resource usage

- Service status

These tools make troubleshooting much easier.

Best Practices

To keep your Dokploy server secure and reliable:

- Enable automatic backups.
- Use SSH keys instead of passwords.
- Keep Ubuntu and Dokploy updated.
- Monitor CPU, RAM, and storage usage.
- Secure your firewall by allowing only required ports.
- Use strong administrator passwords.

Following these practices helps ensure long-term stability.

Common Issues

Docker Installation Problems

Restart the Docker service.

```
sudo systemctl restart docker
```

Firewall Blocking Access

Allow required ports.

```
sudo ufw allow 80
sudo ufw allow 443
sudo ufw allow 3000
```

DNS Not Working

Verify DNS records and allow time for propagation.

Conclusion

Running Dokploy on a VPS is one of the most cost-effective ways to host modern applications while maintaining complete infrastructure control. With Docker-native deployment, Git integration, automatic SSL certificates, and a user-friendly dashboard, Dokploy makes self-hosting significantly easier than traditional server management.

Whether you’re a solo developer, startup founder, freelancer, or agency, deploying Dokploy on a VPS provides the flexibility, scalability, and ownership needed to build reliable hosting environments without recurring platform fees.